

GIT II

Branques	1
Git branch	1
La rama main (master)	2
Crear una rama nueva	2
Pasar de una rama a otra	2
Renombrar una rama	3
Fusionar ramas	3
Subir una rama al repositorio remoto (Github o similares)	3
Bajar ramas remotas en local	4
Borrar una rama	4
Borrado de la rama en local	4
Borrado de la rama en remoto	5
Resolver conflictos en el merge, manualmente	5
TIPS pel projecte	5
Referències	5

Branques

En el día a día del trabajo con Git una de las cosas útiles que podemos hacer es trabajar con ramas. En la práctica permiten que nuestro proyecto pueda tener diversos estados y que los desarrolladores sean capaces de pasar de uno a otro de una manera ágil.

Ramas podrás usar en muchas situaciones. Por ejemplo, imagina que estás trabajando en un proyecto y quieres implementar una nueva funcionalidad en la que sabes que quizás tengas que invertir varios días. Llegará un momento en el que, quizás, aquellos cambios experimentales los quieras subir a producción. Entonces harás un proceso de fusión entre la rama experimental y la rama original, operación que se conoce como **merge** en Git.

Git branch

El comando git branch es el que usaremos principalmente para trabajar con la creación de ramas, borrado de ramas y demás. Sin embargo, no es el único comando para la operativa

GIT: Branques

v1.0

que veremos, ya que existen otros subcomandos de Git útiles y necesarios para trabajar con ramas, como checkout para moverse entre ramas o merge para fusionar ramas.

Puedes comenzar tu primera práctica para trabajar con ramas. Haremos algo tan sencillo como lanzar el comando "git branch" a secas. Esto nos dará el listado de ramas que tengamos en un proyecto. Pero hay que advertir que las ramas de un repositorio local pueden ser distintas de las ramas de un repositorio remoto.

La rama main (master)

Cuando inicializamos un proyecto con Git automáticamente nos encontramos en una rama a la que se denomina "main".

Esta rama es la principal de tu proyecto y a partir de la que podrás crear nuevas ramas cuando lo necesites.

Crear una rama nueva

El procedimiento para crear una nueva rama es bien simple. Usando el comando branch, seguido del nombre de la rama que queremos crear. Vamos a crear la rama

\$ git branch sprint1

Este comando en sí no produce ninguna salida, pero podrías ver las "branches" de un proyecto con el comando "git branch", u obtener una descripción más detallada de las ramas con este otro comando, vamos a probarlo:

\$ git branch

Pasar de una rama a otra

Para moverse entre ramas usamos el comando "git checkout" seguido del nombre de la rama que queremos que sea la activa. Vamos a cambiar a la rama recién creada.

\$ git checkout sprint1

\$ git branch

Esta sencilla operación tiene mucha potencia. Vamos a hacer màgia, porque nos cambiará automáticamente todos los archivos de nuestro proyecto, los de todas las carpetas, para que tengan el contenido en el que se encuentren en la correspondiente rama.

Para saber en qué rama estamos actualmente podemos editar el fichero HEAD de la carpeta oculta .git

.git/HEAD lo podremos comprobar.

De momento en nuestro ejemplo las dos ramas tenían exactamente el mismo contenido, pero ahora podríamos empezar a hacer cambios en el proyecto sprint1 y hacer sus

GIT: Branques

v1.0

correspondientes commit y entonces los archivos tendrán códigos diferentes, de modo que puedas ver que al pasar de una rama a otra hay cambios en los archivos.

El comando checkout tiene la posibilidad de permitirte crear una rama nueva y moverte a ella en un único paso. Para crear una nueva rama y situarte sobre ella tendrás que darle un nombre y usar el parámetro -b. Vamos a probar.

\$ git checkout -b pruebas //creas la rama y te posicionas directamente en ella

Como estás entendiendo, el proyecto puede tener varios 'estados' en un momento dado y tú podrás moverte de uno a otro con total libertad y sin tener que cambiar de carpeta ni nada parecido.

Renombrar una rama

Para renombrar una rama usaremos la siguiente sintaxis.

\$ git branch -m [nombre_actual] [nombre_nuevo]

Prueba a renombrar la otrarama que teníamos a menu_nuevo.

Fusionar ramas

A medida que crees ramas y cambies el estado de las carpetas o archivos tu proyecto empezará a divergir de una rama a otra. Llegará el momento en el que te interese fusionar ramas para poder incorporar el trabajo realizado a la rama master.

El proceso de fusionado se conoce como "merge" y puede llegar a ser muy simple o más complejo si se encuentran cambios que Git no pueda procesar de manera automática.

Git para procesar los merge, usa un antecesor común y comprueba los cambios que se han introducido al proyecto desde entonces, combinando el código de ambas ramas.

Para hacer un merge nos situamos en una rama, en este caso la "master", y decimos con qué otra rama se debe fusionar el código.

Un merge necesita un mensaje, igual que ocurre con los commit

\$ git merge sprint1 -m 'Esto es un merge con mensaje'

Subir una rama al repositorio remoto (Github o similares)

Por mucho que hagas la operativa descrita para crear ramas en tu ordenador, y las puedas ver en tu repositorio local con git branch, las ramas no se publicarán en Github o cualquier otro hosting de repositorios remoto. Para que esto ocurra tienes que realizar específicamente la acción de subir una rama determinada.

GIT: Branques

v1.0

La operativa de publicar una rama en remoto la haces mediante el comando push, indicando la opción "-u" y el nombre de la rama que deseas subir. Por ejemplo de esta manera:

git push -u origin sprint1

Así estamos haciendo un push, empujando hacia origin (que es el nombre que se suele dar al repositorio remoto), la rama con nombre "sprint1".

Bajar ramas remotas en local

Cuando estamos trabajando en un equipo de desarrolladores, muchas veces necesitamos bajar alguna rama nueva del repositorio remoto, creada por otro miembro del equipo. Para no tener que estar pendiente de cuándo alguien crea una nueva rama y de cómo se llama, vamos a ver una forma cómoda de tener nuestro repositorio local siempre actualizado con las nuevas ramas del remoto.

Al hacer un git Branch, solo se muestra la master

Para mostrarnos todas las ramas utiliza:

git branch -all

Si queremos crear las correspondientes ramas locales, sólo tenemos que ir a cada una de las ramas con git checkout:

Borrar una rama

En ocasiones puede ser necesario eliminar una rama del repositorio, por ejemplo porque nos hayamos equivocado en el nombre al crearla. Aquí la operativa puede ser diferente, dependiendo de si hemos subido ya esa rama a remoto o si todavía solamente está en local.

Borrado de la rama en local

Esto lo conseguimos con el comando git branch, solamente que ahora usamos la opción "-d" para indicar que esa rama queremos borrarla.

\$ git branch -d rama_a_borrar

Sin embargo, puede que esta acción no nos funcione porque hayamos hecho cambios que no se hayan salvado en el repositorio remoto, o no se hayan fusionado con otras ramas. En el caso que queramos forzar el borrado de la rama, para eliminarla independientemente de si se ha hecho el push o el merge, tendrás que usar la opción "-D".

\$ git branch -D rama_a_borrar

GIT: Branques

v1.0

i *Importante: Debes prestar especial atención a esta opción "-D", ya que al eliminar de este modo pueden haber cambios que ya no se puedan recuperar.*

Borrado de la rama en remoto

Si la rama que queremos eliminar está en el repositorio remoto, la operativa es un poco diferente. Tenemos que hacer un push, indicando la opción --delete, seguida de la rama que se desea borrar.

```
$git push origin --delete rama_a_borrar
```

Resolver conflictos en el merge, manualmente

[07_GESTIO_GIT II_conflictos]

TIPS pel projecte

1. Al projecte podem tenir 3 branques:
 - Main: L'actualitzarem amb la versió de la Release, un cop finalitzat el Sprint.
 - sprint1: La que farem servir per desenvolupar el sprint1
2. Aneu fent merges sovint per resoldre els conflicte asap (as soon as possible).
3. Recordeu el .gitignore per la carpeta de /node_modules/

Referències

Git Branch

<https://git-scm.com/docs/git-branch>

Create Your First GitHub Pull Request

<https://youtu.be/JM34MokHQSQ>

Qué es Git flow y cómo funciona

<https://cleventy.com/que-es-git-flow-y-como-funciona/#:~:text=Es%20ah%C3%AD%20donde%20entra%20en,a%20los%20lanzamientos%20del%20proyecto.>

Understanding the Git flow

<https://waydev.co/understanding-the-git-flow/>